Ordonnancement des pieces dans un systeme de traitement de surfaces

D. Bolignano, Jean-Marie Proth, K. Voyiatzis

INRIA

CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(3) 954 90 20

Rapports de Recherche

N° 234

ORDONNANCEMENT DES PIÈCES DANS UN SYSTÈME DE TRAITEMENT DE SURFACES

Dominique BOLIGNANO
Jean-Marie PROTH
Kosta VOYIATZIS

Août 1983

ORDONNANCEMENT DES PIECES DANS UN SYSTEME DE TRAITEMENT DE SURFACE.

Dominique BOLIGNANO*

Jean-Marie PROTH**

Kosta VOYIATZIS**

* SODETEG T.A.I., Z.I. de Buc, 283 rue de la Minière, 78530 BUC - Tél. (3) 956.80.60.

** INRIA, Domaine de Voluceau-Rocquencourt, B.P. 105, 78153 LE CHESNAY Cédex - Tél. (3) 954.90.20.

ABSTRACT

In order to perform a surface treatment on a metallic part many consecutive chemical baths have to be done. This is intended to give the parts particular properties. Let us denote by "manufacturing process" the ordered sequence of consecutive baths which constitute a surface treatment. The duration of a particular bath must lie within the two bounds given by the corresponding "manufacturing process". Furthermore, no delay is allowed between two consecutive baths.

The parts to be surfaced arrive randomly into the workshop and must fit into the preexisting schedule which cannot be altered.

In the following paper we present an algorithm giving an optimal solution to this scheduling problem. The number of reiterations performed by this algorithm is bounded by the number of available slots.

RESUME

Appliquer un traitement de surface à une pièce métallique consiste à la plonger successivement dans des bains chimiques différents. Le but de l'opération est de lui donner des propriétés intéressant ses couches superficielles (résistance à la corrosion par exemple). Nous appellerons gamme une suite ordonnée de bains et, pour chacun de ces bains, un temps minimal et un temps maximal de séjour de la pièce concernée. Aucune attente n'est admise entre deux bains successifs.

La pièce à traiter ainsi que la gamme qui la concerne ne sont connus que très peu de temps avant le lancement, et on ne peut remettre en question aucun des ordonnancements précédents. Le problème consiste à choisir les cuves qui seront utilisées et l'instant de début de traitement dans chacune

d'elles afin que le traitement de surface soit terminé dans les meilleurs délais. Nous proposons dans ce papier un algorithme "temps réel" qui répond à la question. Nous présentons un exemple numérique.

I - INTRODUCTION.

Appliquer un traitement de surface à une pièce consiste à la plonger successivement dans des bains de compositions chimiques différentes afin de faire acquérir certaines propriétés à ses couches superficielles. Le temps de séjour d'une pièce dans un bain donné est compris entre deux bornes connues, et le passage d'un bain au bain suivant doit se faire sans délai. Nous appellerons gamme une suite ordonnée de bains avec, pour chacun d'eux, un temps minimal et un temps maximal de séjour.

L'ordonnancement des pièces prises en compte antérieurement ne peut être remis en question à l'arrivée d'un nouvelle pièce.

Un bain chimique donné est éventuellement contenu dans plusieurs cuves différentes. Lorsqu'une pièce parvient dans l'atelier, chaque cuve peut être partiellement retenue pour l'exécution de gammes amorcées précédemment. Une fenêtre est un intervalle de temps durant lequel une cuve est disponible. Ordonnancer une nouvelle pièce consiste à choisir une cuve pour chacun des bains concernés par la gamme et, pour chacune de ces cuves, une fenêtre plus grande ou égale au temps minimal de séjour de la pièce dans le bain. L'ensemble de ces choix doit répondre aux contraintes déjà énoncées et permettre d'exécuter le traitement le plus rapidement possible.

Dans la suite, nous donnons d'abord les notations utilisées. Nous présentons ensuite les propriétés qui sont à la base de l'algorithme.

Nous proposons enfin un exemple numérique.

II - NOTATIONS ET POSITION DU PROBLEME.

Le système de traitement de surface comporte M bains différents. Pour faciliter l'exposé, nous ne considérons que les m bains concernés par la gamme que nous souhaitons exécuter.

Pour $i = 1,2,\ldots,m$, n_i est le nombre de cuves contenant le bain i. Le problème se pose à l'instant t_0. A cet instant, nous connaissons les périodes de disponibilité de chacune des cuves.

Pour $i = 1,2,\ldots,m$ et $j = 1,2,\ldots,n_i$, $a^k_{i,j}$ représente le début de la k^{me} période de disponibilité pour la cuve j appartenant à l'ensemble des cuves qui contiennent le bain $i(k = 1,2,\ldots,f_{i,j})$. De la même manière, $b^k_{i,j}$ représente la fin de cette k^{me} période.

Nous noterons encore θ^g_i $(i = 1,\ldots,m)$ et $\theta^g_i + \delta^g_i$ les bornes inférieure et supérieure de la durée de séjour dans le bain i d'une pièce ayant à subir la gamme g. Pour simplifier les notations, nous n'utiliserons pas l'indice g dans la suite (nous nous intéressons en effet à l'instant t_0 à une seule pièce et aucune confusion n'est possible).

Pour $i = 1,2,\ldots,m$, nous désignons par t_i l'instant (que nous cherchons) où le i^{me} bain débute. Le fait que les bains se succèdent sans interruption entraine que, pour $i = 2,3,\ldots,m$, t_i représente également l'instant où le $(i - 1)^{me}$ bain se termine. t_{m+1} est l'instant de fin du m^{me} bain et donc l'instant de fin de traitement de la pièce.

Nous appelons "k^{me} fenêtre pour la cuve $j \in \{1,2,\ldots,n_i\}$ correspondant au bain $i \in \{1,2,\ldots,m\}$" l'intervalle de temps :

$$(1) \qquad F^k_{i,j} = (a^k_{i,j}, b^k_{i,j})$$

où, pour i et j fixés, la suite $F^k_{i,j}$, $k = 1,\ldots,f_{i,j}$, est classée dans l'ordre croissant des $a^k_{i,j}$ et, pour deux $a^k_{i,j}$ égaux, dans l'ordre décroissant des $b^k_{i,j}$.

En d'autres termes :

$$(F^{k_1}_{i,j} \text{ précède } F^{k_2}_{i,j}) \text{ si et seulement si } ((a^{k_1}_{i,j} < a^{k_2}_{i,j}) \text{ ou}$$

$$(a^{k_1}_{i,j} = a^{k_2}_{i,j} \text{ et } b^{k_1}_{i,j} > b^{k_2}_{i,j}))$$

Nous considérons maintenant les suites :

(2) $$S = \{F^k_{i,j}\}_{i=1,\ldots,m} \text{ avec } j \in \{1,2,\ldots,n_i\} \text{ et } k \in \{1,\ldots,f_{i,j}\}$$

Ces suites sont au nombre de :

(3) $$w = \prod_{i=1}^{m} \sum_{j=1}^{n_i} f_{i,j}$$

Nous écrirons souvent une telle suite :

$$S = (s_i)_{i=1,\ldots,w}$$

Notre problème s'écrit alors :

trouver $t_1 < t_2 < \ldots < t_m < t_{m+1}$ vérifiant, pour une suite de fenêtres $S = \{F^k_{i,j}\}_{i=1,2,\ldots,m}$, $j \in \{1,2,\ldots,n_i\}$ et $k \in \{1,2,\ldots,f_{i,j}\}$:

$$\left.\begin{array}{ll} (4.1) & \theta_i \le t_{i+1} - t_i \le \theta_i + \delta_i \\ (4.2) & t_i \ge a^k_{i,j} \\ (4.3) & t_{i+1} \le b^k_{i,j} \end{array}\right\} \quad \text{pour } i = 1,2,\ldots,m$$

et minimisant t_{m+1}, instant de fin de traitement du produit considéré.

Nous pourrions évidemment utiliser les algorithmes classiques de la programmation linéaire à condition d'accepter de résoudre w problèmes, ce qui est incompatible avec l'exigence "temps réel".

Nous utiliserons, à la place de (4.1), (4.2) et (4.3), les relations suivantes :

$$\left.\begin{array}{ll}(5.1) & t_{i+1} \geq \theta_i + t_i \\ (5.2) & t_i \geq a^k_{i,j} \\ (5.3) & t_{i+1} \leq \theta_i + \delta_i + t_i \\ (5.4) & t_{i+1} \leq b^k_{i,j}\end{array}\right\} \quad \text{pour } i = 1,2,\ldots,m$$

Toute séquence $t_1 < t_2 < \ldots < t_m < t_{m+1}$ qui vérifie ces contraintes pour au moins une suite S de fenêtres est dite admissible. Elle est optimale si, de plus, elle minimise t_{m+1} sur l'ensemble des suites S possibles.

Les résultats qui conduisent à l'algorithme "temps réel" sont donnés dans le paragraphe suivant.

III - RECHERCHE D'UNE SOLUTION OPTIMALE.

Les résultats qui suivent permettent de ne tester qu'un nombre restreint de suites de fenêtres susceptibles de contenir la solution optimale cherchée, à condition de considérer ces suites dans un ordre déduit de celui des fenêtres que nous avons présenté en (1). Nous montrerons que le nombre de suites à considérer est borné supérieurement par le nombre total de fenêtres.

RESULTAT I

Soit S une suite de fenêtres qui n'admet aucune solution admissible. Alors il en est de même pour toute suite S' dont chaque terme est inclus dans le terme correspondant de S.

Démonstration

Soit $S = \{s_i\}_{i=1,\ldots,m}$ avec $s_i = (h_i, H_i)$ pour $i = 1,2,\ldots,m$ et $S' = \{s'_i\}_{i=1,\ldots,m}$ avec $s'_i = (h'_i, H'_i)$ pour $i = 1,2,\ldots,m$ les suites évoquées dans l'énoncé précédent.

Par hypothèse :

(6) $s'_i \subset s_i$ pour $i = 1,2,\dots,m$.

Supposons qu'il existe une suite admissible $T = (t_1,\dots,t_{m+1})$ pour la suite de fenêtres S'. Alors les éléments de T vérifieraient (5.1) à (5.4) (en remplaçant dans (5.2) et (5.4) $a^k_{i,j}$ et $b^k_{i,j}$ respectivement par h'_i et H'_i).

Mais, compte tenu de (6) :

$t_i \geq h'_i$ entraîne $t_i \geq h_i$

et $t_{i+1} \leq H'_i$ entraîne $t_{i+1} \leq H'_i$.

Si bien que T vérifierait (5.1) à (5.4) et serait admissible pour S, ce qui est contraire à l'hypothèse.

D'où le résultat.

□

Pour tout $i \in \{1,2,\dots,m\}$, nous noterons $\{\alpha_{i,l}\}_{l=1,\dots,q_i}$ l'ensemble des $\{a^k_{i,j}\}^{j = 1,\dots,n_i}_{k = 1,\dots,f_{i,j}}$ classées dans l'ordre croissant.

Pour $l = 1,2,\dots,q_i$, $\beta_{i,l}$ désignera l'extrémité de la fenêtre d'origine $\alpha_{i,l}$. q_i est le nombre de fenêtres attachées aux n_i cuves contenant le bain i.

Nous construisons d'abord la suite $T = \{t_1, t_2,\dots,t_m, t_{m+1}\}$ de la manière suivante :

(7.1) $t_1 = \alpha_{1,1}$

(7.2) $t_i = \text{Max}\ (\alpha_{i,1}, \theta_{i-1} + t_{i-1})$ pour $i = 2, 3,\dots,m$

(7.3) $t_{m+1} = \theta_m + t_m$

Nous construisons ensuite $X = \{x_1, x_2, \ldots, x_{m+1}\}$ de la manière suivante :

(8.1) $\quad x_{m+1} = t_{m+1}$

(8.2) $\quad x_i = \text{Max}\ \{t_i,\ x_{i+1} - \theta_i - \delta_i\}$ pour $i = m, m-1, 1$

RESULTAT 2

La suite X définie ci-dessus vérifie (5.1), (5.2) et (5.3) pour la suite $(\alpha_{i,1}, \beta_{i,1})_{i=1,\ldots,m}$ des fenêtres de disponibilité.

Démonstration

a. Nous montrons d'abord que X vérifie (5.1), c'est-à-dire que :

$$x_{i+1} \geq \theta_i + x_i \qquad \text{pour } i = 1,2,\ldots,m$$

La relation (8.2) conduit à :

$$(9) \qquad \begin{aligned} \theta_i + x_i &= \text{Max}\ (t_i + \theta_i,\ x_{i+1} - \delta_i) \\ &\leq \text{Max}\ (t_i + \theta_i,\ x_{i+1}) \qquad \text{pour } i = 1,\ldots,m \end{aligned}$$

Mais (cf. (7.2) et (7.3))

$$t_i + \theta_i \leq t_{i+1} \qquad \text{pour } i = 1,2,\ldots,m$$

et la relation (9) conduit à :

(10) $\quad \theta_i + x_i \leq \text{Max}\ (t_{i+1},\ x_{i+1})$ pour $i = 1,\ldots,m$

Mais (cf. (8.1) et (8.2)) :

$$x_i \geq t_i \qquad \text{pour } i = 1,\ldots,m+1$$

Finalement, (10) se réécrit :

$$\theta_i + x_i \leq x_{i+1} \qquad \text{pour } i = 1,2,\ldots,m$$

b. La suite X vérifie (5.2) pour la suite de fenêtres de disponibilité considérée.

En effet :

$$t_i \geq \alpha_{i,1} \qquad \text{pour } i = 1,2,\ldots,m \qquad \text{(cf. (7.1) et (7.2))}$$

et

$$x_i \geq t_i \qquad \text{pour } i = 1,2,\ldots,m \qquad \text{(cf. (8.2))}$$

Donc :

$$x_i \geq \alpha_{i,1} \qquad \text{pour } i = 1,2,\ldots,m$$

c. Nous montrons enfin que X vérifie (5.3), c'est-à-dire que :

$$x_{i+1} \leq \theta_i + \delta_i + x_i \qquad \text{pour } i = 1,2,\ldots,m$$

Cette relation est une conséquence directe de (8.2).

□

RESULTAT 3

Si, pour $i_1 \in \{1,2,\ldots,m\}$,

(11) $$x_{i_1+1} > \beta_{i_1,1}$$

alors il n'existe aucune suite de fenêtres de disponibilité contenant l'élément $(\alpha_{i_1,1}, \beta_{i_1,1})$ et qui admet une solution admissible.

Démonstration

Considérons la suite de fenêtres $S' = \{\alpha_{i,j_i}, \beta_{i,j_i}\}_{i=1,\ldots,m}$, où $j_i \in \{1,\ldots,q_i\}$ et telle que $\{\alpha_{i_1,j_{i_1}}, \beta_{i_1,j_{i_1}}\} = \{\alpha_{i_1,1}, \beta_{i_1,1}\}$.

Construisons $X' = (x'_1,\ldots, x'_m, x'_{m+1})$ à partir de S' comme X a été construit à partir de S.

La suite T', obtenue à partir de (7.1), (7.2), (7.3) et S', vérifie, en tenant compte du fait que pour tout $i \in \{1,\ldots,m\}$ les $\alpha_{i,j}$ sont classés par ordre croissant :

$$t_i \leq t'_i \qquad \text{pour } i = 1,2,\ldots,m+1$$

Compte tenu de (8.1) et de (8.2), nous déduisons de la relation précédente :

$$x_i \leq x'_i \qquad \text{pour } i = 1,2,\ldots,m+1$$

D'où, en considérant (11) :

$$x'_{i_1+1} > \beta_{i_1,1}$$

et X' n'est pas admissible car il ne vérifie pas la condition (5.4). Or, quelque soit $V = \{v_i\}_{i=1,\ldots,m+1}$ qui vérifie (5.1), (5.2) et (5.3) pour la suite de fenêtres S', nous avons :

$$x'_i \leq v_i \qquad \text{pour } i = 1,2,\ldots,m+1$$

D'où :

$$v_{i_1+1} > \beta_{i_1,1}$$

et V n'est pas admissible, ce qui achève la démonstration. □

Les résultats précédents conduisent à l'algorithme de recherche d'une solution admissible que nous présentons maintenant. Nous montrerons ensuite que cette solution admissible est également optimale.

ALGORITHME

1. Pour chaque bain, classement des fenêtres dans l'ordre croissant de l'instant de début. Nous obtenons ainsi les suites $(\alpha_{i,l}, \beta_{i,l})_{l=1,\dots,q_i}$ pour $i = 1,2,\dots,m$.

2. Construction de la suite X à l'aide des relations de (7.1) à (8.2).

   2.1. si quelque soit $i_1 \in \{1,2,\dots,m\}$, la relation (11) n'est pas vérifiée, alors X est admissible. Fin de processus.

   2.2. sinon, soit $I_1 \subset \{1,2,\dots,m\}$ l'ensemble des indices i_1 tels que (11) soit vérifiée. On élimine l'ensemble des fenêtres $(\alpha_{i_1,1}, \beta_{i_1,1})_{i_1} \in I_1$.

      2.2.1. S'il ne reste aucune fenêtre à l'un des niveaux $i_1 \in I_1$, le problème n'admet aucune solution admissible. Fin de processus.

      2.2.2. Si, quelque soit $i \in \{1,2,\dots,m\}$, il existe au moins une fenêtre, on reprend le processus en 1.

Cet algorithme peut être amélioré à l'aide du résultat 1.

RESULTAT 4

Si l'algorithme précédent conduit à une solution admissible, elle est optimale (i.e. x_{m+1} est l'instant minimal de fin de traitement de la pièce pour l'ensemble des fenêtres de disponibilité observées).

Démonstration

Soit $S = (s_1,s_2,\dots,s_m,s_{m+1})$ une solution admissible.et $X = (x_1,x_2,\dots,x_m,x_{m+1})$ la solution admissible donnée par l'algorithme précédent.

Nous considérons les relations (7.1) à (8.2).

a. Nous montrons d'abord que

(12) $s_i \geq t_i$ pour $i = 1,2,\ldots,m+1$

Nous savons que (cf.(5.2)) :

$$s_1 \geq \alpha_{1,1}$$

et (cf. (7.1)) :

$$t_1 = \alpha_{1,1}$$

donc

$$s_1 \geq t_1$$

De plus, si

$$s_j \geq t_j \quad \text{pour } j = 1,\ldots,i-1 \text{ et } i > 1$$

alors (cf. (7.2)) :

$$s_i = \text{Max } (\alpha_{i,j}, \theta_{i-1} + s_{i-1}) \geq \text{Max } (\alpha_{i,1}, \theta_{i-1} + t_{i-1}) = t_i,$$

ce qui achève la démonstration.

b. Nous montrons maintenant que :

$$s_i \geq x_i \quad \text{pour } i = 1,2,\ldots,m+1$$

D'après (12),

$$s_{m+1} \geq t_{m+1}$$

D'où (cf. (8.1)) :

$$s_{m+1} \geq x_{m+1}$$

De plus, si

$$s_j \geq x_j \quad \text{pour } j = m+1,\ldots,i+1 \ (i > 1)$$

alors (cf. (8.2))

$$x_i = \mathrm{Max}\ (t_i,\ x_{i+1} - \theta_i - \delta_i) \leq \mathrm{Max}\ (t_i,\ s_{i+1} - \theta_i - \delta_i)$$

Mais, S étant admissible :

$$s_{i+1} - \theta_i - \delta_i \leq s_i$$

Finalement :

$$x_i \leq \mathrm{Max}\ (t_i,\ s_i)$$

Mais

$$s_i \geq t_i \quad \text{(cf. (12))}$$

Donc

$$x_i \leq s_i$$

ce qui achève la démonstration.

□

Nous montrons maintenant que le nombre d'itérations à effectuer dans l'algorithme précédent est majoré par le nombre total de fenêtres qui interviennent dans notre problème.

RESULTAT 5

Si f_{ij} est le nombre de fenêtres disponibles pour la cuve j contenant un bain i, si n_i est le nombre de cuves contenant le bain i, et si m est le nombre de bains différents, le nombre d'itérations nécessaires pour exécuter l'algorithme précédent est majoré par :

$$A_{ij} = \sum_{i=1}^{m} \sum_{j=1}^{n_i} f_{ij}$$

Démonstration

Le nombre total de fenêtres est A_{ij}. A chaque itération, soit on obtient la solution optimale X, soit on élimine au moins une fenêtre. D'où le résultat précédent.

□

IV - EXEMPLES.

Nous considérons dix bains différents. Chacun de ces bains est contenu dans trois cuves. Nous travaillons à l'horizon 100. Les longueurs des fenêtres de disponibilité et d'occupation de chaque cuve sont générées au hasard sur [0,12] (répartition uniforme). De même, pour i = 1,2,...,10, les θ_i (temps minimal de séjour dans une cuve contenant le bain i) et les δ_i (écart entre temps minimal et temps maximal de séjour dans une cuve contenant le bain i) sont générés au hasard (loi uniforme) sur (0,9) et (0,1.5) respectivement.

Nous présentons d'abord un exemple dans le détail.

Nous donnons ensuite une répartition du nombre d'itérations divisé par le nombre de fenêtres obtenu à partir de 500 itérations effectuées sur la base des mêmes données que ci-dessus.

IV.1 - Un exemple d'application.

Nous donnons pour chaque cuve de chaque bain les fenêtres disponibles à l'instant considéré. Les fenêtres dans lesquelles se trouveront les éléments de la suite optimale $T = (t_1, t_2, \ldots, t_m, t_{m+1})$ sont soulignées.

Bain numéro 1

Cuve 1 : (0.51-9.45) (<u>11.29-26.10</u>) (31.12-34.34) (38.15-44.18) (56.20-66.06) (66.15-66.47) (72.27-81.46) (86.23-90.11)

Cuve 2 : (3.03-4.22) (10.14-15.73) (24.35-27.89) (37.16-50.90) (57.46-70.65) (75.88-83.86) (92.42-100.00)

Cuve 3 : (7.48-14.62) (23.01-27.14) (34.14-36.13) (39.01-42.25) (51.12-51.46) (60.89-75.68) (80.40-94.08) (99.84-100.00)

Bain numéro 2

Cuve 1 : (2.12-11.36) (15.30-16.24) (24.55-38.62) (47.88-62.58) (74.34-83.83) (95.95-100.00)

Cuve 2 : (<u>11.01-23.31</u>) (28.46-28.64) (39.35-51.39) (56.02-61.37) (68.53-81.10) (87.00-92.77)

Cuve 3 : (13.40-24.86) (32.79-41.10) (55.26-56.44) (64.62-72.11) (83.19-93.95)

Bain numéro 3

Cuve 1 : (2.02-8.75) (<u>17.41-28.42</u>) (40.65-43.18) (46.70-61.46) (62.18-63.35) (70.90-74.23) (84.27-96.72) (98.40-100.00)

Cuve 2 : (2.42-16.34) (19.06-26.19) (29.64-33.31) (43.29-48.65) (57.83-68.20) (76.64-79.75) (88.50-97.43)

Cuve 3 : (5.30-17.13) (29.55-30.03) (40.98-50.39) (57.53-58.12) (59.76-63.43) (75.25-76.96) (86.63-95.53) (96.05-100.00)

<u>Bain numéro 4</u>

Cuve 1 : (2.30-13.66) (20.62-29.95) (39.23-47.52) (54.56-55.01) (59.71-64.18) (69.29-76.60) (84.91-85.15) (95.30-100.00)

Cuve 2 : (2.01-7.97) (<u>20.46-32.57</u>) (46.96-49.18) (55.73-63.61) (68.19-80.92) (93.26-100.00)

Cuve 3 : (13.69-14.19) (22.45-31.65) (42.32-45.44) (57.82-64.49) (76.10-76.62) (85.22-91.42) (98.87-100.00)

<u>Bain numéro 5</u>

Cuve 1 : (14.03-26.27) (35.92-37.01) (40.85-46.63) (50.19-50.83) (64.61-76.80) (83.52-96.01)

Cuve 2 : (14.91-18.99) (19.57-28.91) (41.27-43.00) (43.59-47.24) (61.30-73.69) (80.21-85.03) (88.55-88.57) (97.26-100.00)

Cuve 3 : (<u>14.98-29.72</u>) (36.42-50.11) (54.28-65.44) (68.78-69.43) (73.44-76.23) (86.53-95.72) (99.82-100.00)

<u>Bain numéro 6</u>

Cuve 1 : (1.64-5.86) (10.47-15.75) (21.68-28.61) (44.55-51.23) (64.63-77.02) (84.13-89.95) (97.40-100.00)

Cuve 2 : (1.79-9.24) (14.13-25.89) (32.97-42.63) (53.00-61.54) (69.64-77.74) (89.20-97.71)

Cuve 3 : (2.58-10.89) (<u>18.17-32.02</u>) (36.27-51.17) (54.95-57.27) (68.65-69.28) (89.67-94.62

<u>Bain numéro 7</u>

Cuve 1 : (1.81-9.05) (20.87-27.40) (35.66-46.22) (49.04-50.09) (63.52-69.73) (75.47-79.49) (91.34-92.81)

Cuve 2 : (8.02-20.93) (27.46-40.22) (41.88-46.15) (48.78-59.13) (65.85-72.19) (80.28-86.50) (91.71-99.82)

Cuve 3 : (8.55-13.77) (<u>25.07-36.48</u>) (41.52-48.03) (49.20-55.26) (66.15-78.67) (87.80-96.76)

Bain numéro 8

Cuve 1 : (0.68-11.80) (13.78-20.93) (<u>31.68-42.46</u>) (45.25-57.38) (60.97-70.07) (75.07-89.76) (99.06-100.00)

Cuve 2 : (6.35-11.89) (19.90-24.51) (34.64-42.01) (48.26-48.32) (57.98-67.94) (70.71-76.03) (79.89-87.68) (91.08-98.02)

Cuve 3 : (13.01-22.81) (28.42-30.06) (37.46-43.85) (53.17-64.77) (71.07-83.53) (89.89-98.02)

Bain numéro 9

Cuve 1 : (2.36-4.53) (17.07-31.33) (<u>38.22-44.73</u>) (49.66-64.03) (65.90-78.75) (81.79-96.16)

Cuve 2 : (10.59-16.91) (26.66-30.80) (39.77-44.41) (52.18-54.61) (66.80-69.20) (77.19-88.81) (90.93-96.17)

Cuve 3 : (12.21-16.89) (29.41-34.44) (41.67-42.24) (54.07-57.71) (68.10-78.82) (89.35-90.14) (99.03-100.00)

Bain numéro 10

Cuve 1 : (2.93-4.94) (7.59-10.14) (22.66-25.83) (34.27-38.29) (<u>44.03-52.68</u>) (56.68-60.98) (64.33-67.42) (74.58-88.64) (89.62-97.62)

Cuve 2 : (4.07-6.56) (11.41-17.79) (25.56-36.56) (43.51-44.63) (48.63-55.13) (68.45-72.34) (86.69-100.00)

Cuve 3 : (5.73-14.99) (19.37-34.05) (39.68-53.07) (62.36-74.18) (86.05-96.06)

Le tableau suivant fournit la gamme à exécuter :

Bain	Temps minimal de séjour	Ecart entre les temps mini et maxi de séjour
1	2.51	1.29
2	2.86	0.20
3	1.27	0.44
4	3.70	0.58
5	0.42	0.17
6	0.63	1.37
7	4.08	1.30
8	7.93	1.04
9	4.63	0.43
10	3.63	1.33

La solution optimale finalement obtenue est la suivante :
11.89, 15.69, 18.75, 20.46, 24.16, 24.58, 26.30, 31.68, 39.61, 44.24, 47.87.

IV.2 - Des résultats statistiques à partir des données précédentes.

Sur 500 simulations effectuées comme indiqué précédemment, 357 ont conduit à des situations qui n'admettaient aucune solution admissible.

Le rapport du nombre d'itérations sur le nombre de fenêtes est inférieur à 0.1 dans 121 cas et compris entre 0.1 et 0.2 dans 22 autres cas. Ces expériences, ainsi que celles que nous avons menées par ailleurs, montrent que la majoration proposée par le résultat 5 est très large dans la majorité des cas où le système admet une solution admissible.

V - CONCLUSION.

L'algorithme que nous proposons est incontestablement d'exécution rapide. Il a été mis en place dans l'industrie par les ingénieurs de la SODETEG-T.A.I., sous une forme légèrement plus complète et qui tient compte des transports de pièces entre les différentes cuves.

Bien entendu, le résultat fourni est un optimum local : on ne cherche pas à réorganiser le passé, et on ne fait aucune hypothèse sur l'avenir. Cette manière de procéder se justifie par le fait que les délais sont excessivement courts, ce qui exige que les pièces soient lancées dès leur arrivée. L'expérience montre en outre que l'algorithme que nous venons de présenter laisse peu de fenêtres libres : il explore en effet en priorité les fenêtres les moins éloignées dans le temps, lesquelles sont également les fenêtres les plus malaisées à utiliser du fait de leur faible taille.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique